The Topic Detection and Tracking Phase 2 (TDT2) Evaluation Plan

Introduction

The purpose of the TDT2 project is to advance the state of the art in Topic Detection and Tracking. The general TDT task domain is to be explored and technology is to be developed in the context of an evaluation-driven R&D paradigm, in which key technical challenges are defined and supported by formal evaluations. This document presents these formal task definitions and the performance measures and procedures to be used to direct the research and evaluate technical capabilities and research progress.

The TDT2 project addresses multiple sources of information, including both text and speech. These sources are namely newswires and radio and television news broadcast programs. The information flowing from each source is modeled as a sequence of stories (and non-stories). These stories provide information on many topics. The technical challenge is to identify and to follow the topics being discussed in these stories.

Topics

In the initial TDT study, conducted during 1996 and 1997, techniques were explored for detecting the appearance of new topics and for tracking the reappearance and evolution of them. Early on in this study, the notion of a topic was modified and sharpened to be an "event", meaning something that happens at some specific time and place. For example, the eruption of Mount Pinatubo on June 15th, 1991 is considered to be an event, whereas volcanic eruption in general is not. Events might be unexpected, such as an airplane crash, or expected, such as a political election.

In the TDT2 project, the notion of a topic as an event has been broadened. How to define "topic" is a fundamental issue, of the greatest importance. It is also a very difficult problem, one which has not been fully resolved and for which no perfect solution exists. However, for the purposes of the TDT2 project:

A topic is defined to be a seminal event or activity, along with all directly related events and activities.

Stories will be considered to be "on topic" whenever the story is *directly* connected to the associated event. So, for example, a story on the search for survivors of an airplane crash, or on the funeral of the crash victims, will be considered to be a story on the crash event. (This is different from the TDT pilot study, where consequential events were considered to be separate events.) Obviously there must be limits to this inclusiveness. (For example, stories on FAA repair directives that derive from a crash investigation probably would not be considered to be stories on the crash event.) As part of this effort to broaden the notion of a topic, topics will also include coherent and topical news foci, even when there is no clear underlying event.

The Corpus

A corpus of text and transcribed speech is being developed to support the TDT2 project. This corpus will span the first half of 1998, January through June, and will include approximately 60,000 stories. There will be 6 sources, including 2 newswires, 2 radio programs and 2 television programs.[1] The corpus will include transcriptions of the radio and TV stories in addition to recordings of the audio signal. Two distinct transcriptions of the audio sources will be provided. These are namely a manual transcription (produced using closed captioning and program transcription data) and an automatic transcription (provided by

[1] For newswire sources each story is clearly delimited by the newswire format. For radio and TV sources, however, the segmentation of the broadcast audio into stories is not so clear. Segmentation of audio sources will be performed with an eye toward the TDT task domain, and as a guide, audio sources will be segmented into stories so that each "story" discusses a single topic. It turns out that closed captioning services provide just such a kind of segmentation, and therefore closed captioning practice will be followed.

Dragon Systems). The transcriptions will be annotated and provided in SGML format. Segment boundary times will be provided for the audio sources, and the resulting segments will be labeled according to whether they are judged as “stories” or “non-stories”.

Only a sub-sample of data from each source will be included in the TDT2 corpus. Each of these samples will be a continuous recording of approximately 20 contiguous stories for newswire sources and of 30- or 60-minute broadcasts for audio sources. Each of these recordings will be stored in a separate file. (Audio broadcasts will be subdivided, if necessary, into recordings of 30-minute duration, to limit the duration of any specific recording to be no more than 30 minutes.)

A set of approximately 100 target topics will be identified to support the TDT2 research effort. These topics will span the period of data collection uniformly and will also span a spectrum of topic types. Candidate topics will be determined from stories selected at random from the corpus. Each target topic will be explicitly represented in a formal five-part definition:

1. A title, to serve as a mnemonic handle and efficient reference
2. A three-part identification of the seminal event – what/where/when.
3. A link to the story from whence the topic was derived.
4. A reference to the principle(s) of interpretation used to delimit the topic.[2]
5. An explicit description and summary of the topic.

The TDT2 corpus will be completely annotated with respect to these topics, so that each story in the corpus is appropriately tagged for each target topic it discusses. These story-topic tags (Tag [story, topic]) assume a value of YES if the story discusses the target topic, or BRIEF if that discussion comprises less than 10% of the story. Otherwise the (default) tag value is NO.

The TDT2 corpus will be divided into three parts for research management purposes. The first third of the corpus (i.e., the data collected in Jan/Feb 1998) will comprise the *training* set. This data may be used without limit for research purposes. The middle third of the corpus (Mar/Apr 1998) will comprise the *development test* set. This data will be freely available for testing TDT algorithms, but its use should be restricted to diagnostic purposes (rather than direct corpus-based training purposes). The last third of the corpus (May/Jun 1998) will comprise the *evaluation test* set. This data will be reserved for final formal evaluation of performance on the TDT tasks at the end of 1998.

The Linguistic Data Consortium (LDC)[3] is preparing the TDT2 corpus and will make it available to the TDT2 research participants in phases, as early as possible, in order to accelerate the research effort. The corpus will also be made available to the research community at large when it is completed. More detailed information about the TDT2 corpus may be obtained at *http://www.ldc.upenn.edu/TDT/*.

The Tasks

The TDT2 project is concerned with the detection and tracking of topics. The input to this process is a stream of stories. This stream may or may not be pre-segmented into stories, and the topics may or may not be known to the system (i.e., the system may or may not be trained to recognize specific topics). This leads to the definition of three technical tasks to be addressed. These are namely the segmentation of a news source into stories, the tracking of known topics, and the detection of unknown topics.

[2] Principles of interpretation were created to help determine the limits of TDT2 topics. This boundary determination is sometimes extremely difficult and arbitrary. The question is where to draw the line on including (or excluding) “related” events. The difficulty of this task was eased considerably by identifying certain general types of topics and creating specific boundary determination rules for each of those topic types. These rules are contained in a document called the *TDT2 Principles of Interpretation*.

[3] The Linguistic Data Consortium
3615 Market Street
Suite 200
Philadelphia, PA, 19104-2608, USA.

Phone: 215/898-0464
Fax: 215/573-2175
ldc@ldc.upenn.edu
http://www.ldc.upenn.edu

The Story Segmentation Task

The story segmentation task is defined to be the task of segmenting the stream of data from a source into its constituent stories. Since text (newswire) sources are supplied in segmented form, this task applies only to audio (radio and TV) sources. Segmentation of audio signals may be performed directly on the audio signal itself or on the various textual transcriptions of the audio signal.

The Topic Tracking Task

The topic tracking task is defined to be the task of associating incoming stories with topics that are known to the system. A topic is "known" by its association with stories that discuss the topic. Thus each target topic is defined by one or more stories that discuss it. To support this task, a set of training stories is identified for each topic to be tracked. The system may train on the target topic by using all of the stories in the corpus, up through the most recent training story. The tracking task is then to correctly classify all subsequent stories as to whether or not they discuss the target topic.

The Topic Detection Task

The topic detection task is defined to be the task of detecting and tracking topics not previously known to the system. It is characterized by a lack of knowledge of the topic to be detected. Therefore the system must embody an understanding of what a topic is, and this understanding must be *independent of topic specifics*. In the topic detection task, the system must detect new topics as the incoming stories are processed.[4] The system must then proceed to associate input stories with those topics. Thus this process identifies a set of topics, as defined by their association with the stories that discuss them.

The Evaluation

To assess TDT application potential, and to calibrate and guide TDT technology development, TDT task performance will be evaluated formally according to a set of rules for each of the three TDT tasks. The general approach to evaluation will be in terms of classical detection theory, in which performance is characterized in terms of type I and type II errors. Type I errors are "misses", meaning that the target is not detected when it is present. Type II errors are "false alarms", meaning that the target is falsely detected when it is not present. In this framework, different topics will be treated independently of each other, and a system will have separate outputs for each of the target topics.

All sources must be processed together, and processing must be in chronological order. Source file sequencing will be controlled by means of a simple list of chronologically ordered source file names. Each source file will contain an uninterrupted recording of contiguous source data, and it will be assumed that there is no temporal overlap between different source files.[5]

For all three tasks, allowable information for the system to use will include knowledge of the source of the data and knowledge of the time of the stories. Specific evaluation rules for the three tasks now follow.

The Story Segmentation Task

Segmentation algorithms will be evaluated in terms of their ability to correctly locate the boundaries between stories. Only audio sources will be evaluated. Each file of source data is processed separately. The source data file will be either a transcribed version of the audio source, in the form of an untagged text stream file, or it will be the original audio data file, depending on the selected condition

Segmentation output must be performed as the data is being processed, but a certain amount of look-ahead will be allowed. This deferral period is a primary task parameter. (Longer deferral periods presumably will provide better segmentation performance.)

[4] Decisions may be deferred for a limited period, however.

[5] It is unlikely that data in different source files will never overlap. This assumption is made, however, because of the great simplification provided, and judging any temporal disruption as minor because of the limited time duration represented by each source file.

A primary task parameter will be this deferral period, N_d. For source data in text form, N_d is the number of words of deferral allowed. For source data in original audio form, N_d is the number of seconds of deferral allowed. The values to be used in the TDT2 project are 100, 1000 and 10,000 words for textual source data, and 30, 300 and 3,000 seconds for audio source data.[6]

The segmentation test will be directed by an index file containing a list of source files to be segmented. Systems must process the source files in order of occurrence. For the segmentation task, there will be three different index files, corresponding to audio data that is manually transcribed, automatically transcribed, and untranscribed. Each index file will be structured as follows:[7]

The first data record in the index file will contain

```
#         Task    PointerType
```

where

Task is an indication of the TDT task to be run. In this case the field will contain 'SEGMENTATION'.

PointerType is the type of boundaries to be output by the system. The possible values are 'RECID' for segmentation of source data in text form, or 'TIME' for segmentation performed directly on the audio signal.

Each subsequent data record in the index file will identify a source file to process. These records will have only field:

```
Source_file
```

where

Source_file is the file name of the source data file to be processed. The source data file is either an untagged text stream file, or the original audio data file, depending on the selected condition.

Source_file is the filename of the source file being processed.

Segmentation systems under evaluation must record segmentation decisions in an output file, one record for each hypothesized story boundary. The first record in this file will contain three fields that specify information that applies globally to the whole file. These 3 fields will contain:

System N_d PointerType

where

System is an alphanumeric character string that uniquely identifies the system being tested. (E.g., CDM_P05-8.v37)

N_d is the allowed deferral period, in words for text files and in seconds for audio files.

PointerType is the type of boundary indicated in the balance of the file. The possible values are 'RECID' or 'TIME'.

[6] Decisions may not be deferred beyond the end of the current source file, however. This is unnecessary in any case, because source files always begin and end on a story (or non-story) boundary.

[7] As a general convention for TDT2 files, the appearance of a # character as the first character in a record is taken to signal a comment. In addition, for TDT2 system output files, a comment in the first record of a file will be used as a title record. This title record will be displayed along with the evaluation results for that file. Other than this special use as a title, the remainder of a record will be ignored whenever a # character is encountered.

Each subsequent data record in the file will identify a hypothesized boundary. These records will have two fields and will contain:

Source_file   Boundary

where

Source_file is the filename of the source file being processed.

Boundary is a hypothesized boundary. For text files, Boundary is the index number of the first word in the hypothesized segment, in the range {1, 2, . . .}. For audio files, Boundary is the time of the beginning of the segment, in the range {0.0, . . .}. (It isn't necessary to output the beginning of the first segment.) The hypothesized Boundary points must occur in chronological order.

Segmentation performance will be measured using a modified version of an error metric suggested by John Lafferty.[8] This method avoids dealing with boundaries explicitly by measuring the probability that two sentences are correctly classified as to whether they belong to the same story.

Three modifications have been made to Lafferty's error measure:

1. The unit of distance has been changed from the sentences that Lafferty used. TDT will use word indices for text sources and time (in seconds) for audio sources.

2. The boundary test (as to whether the two sentences/words/times belong to the same story) is made at a fixed distance rather than a probabilistic distance.

3. The error measure is split into miss and false alarm probabilities, so as to represent and evaluate the segmentation task as a formal detection task.

For source data in text form, distances will be measured in terms of word indices, and the boundary test will be made at a separation of *k* words. Choice of *k* is a critical consideration in order to produce a meaningful and sensitive evaluation. For the TDT2 project, *k* will be 50 words, and the miss and false alarm probabilities will be computed as:

$$\mathbf{P}_{\mathbf{Miss}} = \sum_{s \in \{seg\}} \left\{ \sum_{i=1}^{N_s - k} \left\{ \mathrm{d}_{hyp_s}(i, i+k) \cdot \left(1 - \mathrm{d}_{ref_s}(i, i+k)\right) \right\} \right\} \Bigg/ \sum_{s \in \{seg\}} \left\{ \sum_{i=1}^{N_s - k} \left(1 - \mathrm{d}_{ref_s}(i, i+k)\right) \right\}$$

$$\mathbf{P}_{\mathbf{False\,Alarm}} = \sum_{s \in \{seg\}} \left\{ \sum_{i=1}^{N_s - k} \left\{ \left(1 - \mathrm{d}_{hyp_s}(i, i+k)\right) \cdot \mathrm{d}_{ref_s}(i, i+k) \right\} \right\} \Bigg/ \sum_{s \in \{seg\}} \left\{ \sum_{i=1}^{N_s - k} \mathrm{d}_{ref_s}(i, i+k) \right\}$$

where the summation is over all the words in the stories of all the source files in the test corpus and where

$$\mathrm{d}_{sys}(i, j) = \begin{cases} 1 & \text{when words } i \text{ and } j \text{ are deemed by } sys \text{ to be within the same story} \\ 0 & \text{otherwise} \end{cases}$$

For source data in audio form, distances will be measured in terms of time (seconds), and the boundary test will be made at a separation of Δ seconds. Choice of Δ is a critical consideration in order to produce a meaningful and sensitive evaluation. For the TDT2 project, Δ will be chosen to be 15 seconds, and the miss and false alarm probabilities will be computed as:

[8] "Text Segmentation Using Exponential Models", by Doug Beeferman, Adam Berger, and John Lafferty.

$$\mathbf{P_{Miss}} \;=\; \sum_{s\in\{seg\}} \left\{ \int_{t=0}^{T_s-\Delta} \left\{ d_{hyp_s}(t,t+\Delta)\cdot\left(1-d_{ref_s}(t,t+\Delta)\right) \right\} \right\} \Bigg/ \sum_{s\in\{seg\}} \left\{ \int_{t=0}^{T_s-\Delta} \left(1-d_{ref_s}(t,t+\Delta)\right) \right\}$$

$$\mathbf{P_{FalseAlarm}} \;=\; \sum_{s\in\{seg\}} \left\{ \int_{t=0}^{T_s-\Delta} \left\{ \left(1-d_{hyp_s}(t,t+\Delta)\right)\cdot d_{ref_s}(t,t+\Delta) \right\} \right\} \Bigg/ \sum_{s\in\{seg\}} \left\{ \int_{t=0}^{T_s-\Delta} d_{ref_s}(t,t+\Delta) \right\}$$

where the integration is over the entire duration of all the stories in the source files in the test corpus[9] and where

$$d_{sys}(t_1,t_2) \;=\; \begin{cases} 1 & \text{when times } t_1 \text{ and } t_2 \text{ are deemed by } sys \text{ to be within the same story} \\ 0 & \text{otherwise} \end{cases}$$

A segmentation cost function, C_{Seg}, will be used to combine the miss and false alarm probabilities into a single evaluation score. C_{Seg} will serve as a bottom-line performance measure, to help in directing research effort and measuring research progress:

$$C_{Seg} \;=\; C_{Miss}\cdot P_{Miss}\cdot P_{seg} \;+\; C_{FalseAlarm}\cdot P_{FalseAlarm}\cdot(1-P_{seg})$$

where

C_{Miss} (the cost of a miss) = 10.

$C_{FalseAlarm}$ (the cost of a false alarm) = 1.

P_{seg} (the *a priori* probability of a segment being within an interval of k words or Δ seconds) = 0.1. (P_{seg} was chosen based on the TDT2 training corpus.)

In summary, the evaluation of story segmentation may be conducted under a total of up to 9 different conditions. This number is the product of 3 source conditions and 3 deferral periods:

- Three source conditions:
  - ◊ audio – **manual transcription**
  - ◊ audio – **automatic transcription**
  - ◊ audio – **sampled data signal**
- Three segmentation decision deferral periods:

For source data in text form:	**100**	**1000**	**10,000**	words
For source data in audio form:	**30**	**300**	**3,000**	seconds

Each research site is encouraged to study as many of these conditions as may be productively done. However, all sites that perform the segmentation evaluation are required to perform at least one evaluation under common default conditions. The default conditions are:

- ◊ source condition – **automatic transcription**
- ◊ segmentation deferral period – **10,000** words

The Topic Tracking Task

Each topic is to be treated separately and independently. In training the system for any particular target topic, allowable information includes the training set and topic tags *for that target topic only*. During the evaluation of each target topic, no information is given on any other topic.

[9] There exist *non-stories* in the source data in addition to the stories of primary interest. (The most notable example of non-stories is commercials.) To improve the focus and limit the scope of TDT2 research, non-stories will be excluded from the scoring of segmentation performance. This will be done be excluding from the computation of segmentation error probabilities all those regions (*i*,*i*+*k*) or (*t*,*t*+D) that lie wholly within non-stories.

A primary task parameter is the number of stories used to define ("train") the target topic, N_t. N_t is the number of stories tagged YES for the target topic. Evaluation will be conducted for five values of N_t, namely {1, 2, 4, 8, 16}. Thus the maximum value of N_t, $(N_t)_{Max}$, will be 16.

The evaluation corpus will be divided into a different training set and test set for each target topic. The training set will comprise the first part of the corpus, up through the $(N_t)_{Max}$th story tagged YES for the target topic. The test set will comprise the remainder of the corpus that follows. In training, it is the *last* N_t stories tagged YES for the target topic that are to be used. All preceding stories tagged YES (and *all* stories tagged BRIEF) for the target topic are to be excluded from training.

Topic Tracking will be performed under two contrasting conditions. These are namely that 1) story boundaries are given, and 2) story boundaries are *not* given. (This will apply only to the test data. Story boundaries will always be supplied for training data.)

The tracking test will be directed by a set of index files, one per test topic. Each index file will contain a list of topic training stories, followed by a list of non-topic training stories, followed by a list of source files for which the target topic is to be tracked. The index file structure is as follows. First comes:

```
#        Task              PointerType          Topic=N
```

where

Task indicates the TDT task to be run. In this case the field will contain 'TRACKING'.

PointerType is the type of boundaries to be output by the system. The possible values are 'RECID' for text stream segmentation or 'TIME' for audio segmentation.

Topic=N declares the topic id number for which the test is to be run. (This is for documentation only. Allowable information for topic training is restricted to the indices of the training stories that follow.)

Then come $(N_t)_{Max}$ records containing the topic training stories, in chronological order: (Only the last N_t of these stories are to be used for training.)

```
# Topic_training_story     Story_ID    Source_file    Begin       End
```

where

Story_ID is a character string story identifier. (This is the TDT2 corpus "docno".)

Source_file is the file name of the source data file containing the training story. The source data file is either an untagged text stream file, or the original audio data file, depending on the selected condition and the source.

Begin is the word index (or time, if the source file contains sampled audio data) of the beginning of the story.

End is the word index (or time, if the source file contains sampled audio data) of the end of the story.

Then come all the records containing the *non*-topic training stories, in chronological order: (All of these stories may be used for training.)

```
# Non_topic_training_story  Story_ID    Source_file    Begin       End
```

Finally come the records that identify the source files to process. These records will have two fields:

Source_file Begin

where

Source_file is the filename of the source file to be processed.

Begin is the word index (or time, if the source file contains sampled audio data) of the point in the source at which processing is to begin.[10]

The Topic Tracking task is to hypothesize points in the source stream where the target topic is discussed. Topic tracking systems will perform this task by outputting information about these hypothesized points to a file, one record for each putative discussion of the target topic. The first record in this file will contain five fields that specify information that applies globally to the whole file. These 5 fields will contain:

System Boundaries N_t Topic PointerType

where

System is an alphanumeric character string that uniquely identifies the system being tested. (E.g., CDM_P05-8.v37)

Boundaries is either YES or NO, where YES indicates that story boundaries are supplied to the system being tested and NO indicates that they are not.

N_t is the number of stories used to train the system to the target topic.

Topic is an index number in the range {1, 2, . . . ~100} which indicates the target topic being tracked.

PointerType is the type of boundaries to be output by the system. The possible values are 'RECID' for textual source data or 'TIME' for source data in audio form.

Each subsequent data record in the file will identify the beginning point in the source stream of a judgment about whether the target topic is being discussed, along with an associated decision and confidence. This decision and confidence will apply to all subsequent source data until the point specified by the next output data record. These records will have 4 fields and will contain:

Source_file Pointer Decision Score

where

Source_file is the filename of the source file being processed.

Pointer indicates where in the source file the subject discussion commences. For textual source data, Pointer is the index number of the specified word, in the concatenation of all story texts for the source file (in the range {1, 2, . . .}). For source data in audio form, Pointer is the specified time, in seconds.

Decision is either YES or NO, where YES indicates that the system believes that the source being processed does in fact discuss the target topic. NO indicates not.

Score is a real number that indicates how confident the system is that the source being processed discusses the associated topic. More positive values indicate greater confidence.

Before tracking performance may be evaluated, the system output information must be associated with the stories being evaluated. The evaluation software performs this function by first assigning Decision and Score to each word (or time) in the source stream, as specified by the system output. Each story Decision is then computed as the majority decision over all words (or time) in that story, and each story Score is computed as the average score over all words (time) in that story.

[10] Topic tracking begins immediately after the last topic training story. Thus processing begins at the beginning of the source file, except when the source file contains training data.

The topic tracking system may adapt to the test data as it is processed, but only in an unsupervised mode. Supervised feedback is not allowed. (Evaluating over a set of N_t's provides essentially equivalent information.)

Topic tracking decisions must be made by the end of the current source file. Decisions may be deferred to the end of the source data file being processed, but no further. The topic tracking performance metric will be miss and false alarm probabilities over all target topics formulated as follows:

$$P_{Miss} = \sum_{t \in Topics} \left\{ \sum_{s \in Stories_t} \left\{ \left(1 - \mathrm{d}_{hyp}(t,s)\right) \cdot \mathrm{d}_{ref}(t,s) \right\} \right\} \Bigg/ \sum_{t \in Topics} \left\{ \sum_{s \in Stories_t} \mathrm{d}_{ref}(t,s) \right\}$$

$$P_{FalseAlarm} = \sum_{t \in Topics} \left\{ \sum_{s \in Stories_t} \left\{ \mathrm{d}_{hyp}(t,s) \cdot (1 - \mathrm{d}_{ref}(t,s)) \right\} \right\} \Bigg/ \sum_{t \in Topics} \left\{ \sum_{s \in Stories_t} \left(1 - \mathrm{d}_{ref}(t,s)\right) \right\}$$

Where the summation is over all topics, and where

$$\mathrm{d}_{sys}(t,s) = \begin{cases} 1 & \text{if } sys \text{ deemed that topic } t \text{ was discussed in story } s \\ 0 & \text{otherwise} \end{cases}$$

$$Stories_t = \text{all stories in the test corpus after the last training story for topic } t$$

In calculating performance, those stories tagged as BRIEF for the target topic will be excluded from the score computation.

A topic tracking cost function, C_{Track}, will be used to combine the miss and false alarm probabilities into a single evaluation score. C_{Track} will serve as a bottom-line performance measure, to help in directing research effort and measuring research progress:

$$C_{Track} = C_{Miss} \cdot P_{Miss} \cdot P_{topic} + C_{FalseAlarm} \cdot P_{FalseAlarm} \cdot (1 - P_{topic})$$

where

C_{Miss} (the cost of a miss) = 10.

$C_{FalseAlarm}$ (the cost of a false alarm) = 1.

P_{topic} (the *a priori* probability of a story being on some given topic) = 0.005. (P_{topic} was chosen using the TDT2 training topics and training corpus.)

In summary, the evaluation of topic tracking may be conducted under a total of up to 30 different conditions. This number is the product of 3 source conditions, 2 boundary conditions, and 5 training conditions:

- ♦ Three source conditions:
  - ◊ **newswire text** and a **manual transcription** of the audio sources
  - ◊ **newswire text** and an **automatic transcription** of the audio sources
  - ◊ **newswire text** and the **sampled data signal** representing the audio sources[11]

[11] Use of the sampled data signal allows simultaneous use of derivative forms of it, including automatically generated transcriptions.

- ♦ Two story boundaries conditions:
  Given **Not given**

- ♦ Five different training conditions (# of training stories):
  1 **2** **4** **8** **16**

Each research site is encouraged to study as many of these conditions as may be productively done. However, all sites that perform the tracking evaluation are required to perform at least one evaluation under the default conditions. The default conditions are:

- ◊ source – **newswire text** and an **automatic transcription** of the audio
- ◊ boundaries – **Given**
- ◊ training stories – **4**

The Topic Detection Task

The topic detection task is simply to associate together the stories that discuss each topic. Topic detection will use a whole (2-month) sub-corpus as input, and detection is to be performed without knowledge of prior stories and/or topics.[12]

Detection performance will be evaluated only on those stories which discuss one of the predefined target topics and, therefore, only on those stories for which “truth” is known. It is assumed that each story discusses at most one topic.[13] Stories that are tagged as discussing more than one of the target topics will not be used in computing topic detection performance.[14]

The topic detection system must process the input source data in chronological order (according to the test index files). However, the topic detection system is allowed to defer its identification of topics (and its association of stories with them) until a certain amount of subsequent source data is processed. This deferral period is a primary task parameter. (The greater the deferral, the better will be decisions regarding both the detection and identification of new topics and the association of stories with them.)

The topic detection test will be directed by an index file containing a list of source files for which topics are to be detected and tracked. Systems must process the source files in order of occurrence. The index files will follow this format:

```
#          Task    PointerType
```

where

Task is an indication of the TDT task to be run. In this case the field will contain ‘DETECTION’.

PointerType is the type of boundaries to be output by the system. The possible values are ‘RECID’ for source data in text form or ‘TIME’ for audio data.

Each subsequent data record in the file will identify a source file to process. These records will have only one field:

[12] Prior information in the corpus will be available for research, of course, and is expected to be used in developing topic detection technology and systems. Explicit use of stories during testing, however, to build a topic history or other representations or abstractions of the actual story stream, is not allowed.

[13] The assumption that each story discusses only one topic is reasonable for the vast majority of stories. It is being used because it simplifies the task and the evaluation. Those few stories tagged as discussing more than one target topic will be excluded from score computation (but *not* from the evaluation corpus).

[14] While stories that are positively tagged for more than one of the target topics will not be used in computing topic detection performance, such stories will, however, be retained as part of the story stream and will not be eliminated from the corpus during topic detection processing.

Source_file

where

Source_file is the filename of the source data file being processed.

A primary task parameter will be this deferral period, N_f. This is the number of source files, including the source file being processed, for which processing may be completed before committing to an association of a story in the current file with a topic. Evaluation will be conducted for three values of N_f, namely 1, 10 and 100.

Topic Detection will be performed under two conditions. These are namely that 1) story boundaries are given, and 2) story boundaries are not given.

The Topic Detection task is to detect topics and then to hypothesize points in the source stream where they are discussed. Topic Detection systems will perform this task by recording information about these hypothesized points in a file, one record for each putative discussion of a topic, written in ASCII format. The first record in this file will contain four fields that specify information that applies globally to the whole file. These four fields will contain:

System Boundaries N_f PointerType

where

System is an alphanumeric character string that uniquely identifies the system being tested. (E.g., CDM_P05-8.v37)

Boundaries is either YES or NO, where YES indicates that story boundaries are supplied to the system being tested and NO indicates that they are not.

N_f is the deferral period allowed before a decision must be made.

PointerType is the type of boundaries to be output by the system. The possible values are 'RECID' for text stream segmentation or 'TIME' for audio segmentation.

Each subsequent data record in the file will identify a topic, the point in the source stream that discusses it, and a measure of the confidence in the identification. These records will have 5 fields and will contain:

Topic Source_file Pointer Decision Score

where

Topic is an index number in the range {1, 2, . . .} which uniquely indicates the topic.

Source_file is the filename of the source data file being processed.

Pointer indicates where in the source file the subject discussion commences. For textual source data, Pointer is the index number of the specified word, in the concatenation of all story texts for the source file (in the range {1, 2, . . .}). For source data in audio form, Pointer is the specified time, in seconds.

Decision is either YES or NO, where YES indicates that the system believes that the source being processed does in fact discuss the target topic. NO indicates not.

Score is a real number that indicates how confident the system is that the source being processed discusses the associated topic. More positive values indicate greater confidence.

Before detection performance may be evaluated, the system output information must be associated with a story. The evaluation system performs this function by first assigning Decision and Score to each word (or time) in the source stream, for the assigned topic, as specified by the system output. The topic assignment for a story will then be that topic with the greatest number of words (or time) in the story assigned to it. In cases of ties, the topic with the greatest accumulated score will be used.

Topic detection performance will be evaluated by measuring how well the stories belonging to each of the target topics match the stories that the system has assigned to the corresponding system-defined topics. This need for a correspondence relationship presents a problem, however, because no correspondence is given between target topics and system-defined topics. To solve this problem, each reference target topic will be mapped to the one system-defined topic that it best matches. The best match will be the match with the lowest detection cost, where the detection cost is defined as:

$$C_{Det}(R,H) \;=\; C_{Miss} \cdot P_{Miss}(R,H) \cdot P_{topic} \;+\; C_{FalseAlarm} \cdot P_{FalseAlarm}(R,H) \cdot (1 - P_{topic})$$

where

C_{Miss} (the cost of a miss) = 10.

$C_{FalseAlarm}$ (the cost of a false alarm) = 1.

$$P_{Miss}(R,H) \;=\; N_{Miss}(R,H)/|R|$$
$$P_{FalseAlarm}(R,H) \;=\; N_{FalseAlarm}(R,H)/|S|$$

R is the set of stories in a reference target topic.

H is the set of stories in a system-defined topic.

P_{topic} (the *a priori* probability of a story being on some given topic) = 0.005. (P_{topic} was chosen using the TDT2 training topics and training corpus.)

and where

$N_{Miss}(R,H)$ is the number of stories in R that are not in H.

$N_{FalseAlarm}(R,H)$ is the number of stories in H that are not in R.

$|X|$ is the number of elements in set X.

S is the set of stories in the evaluation corpus being processed.

Thus the mapping task is to determine $H(R)$ for all reference target topics, where

$$H(R) = \underset{H}{\operatorname{argmin}}\{C_{Det}(R,H)\}$$

Topic detection performance will be measured in terms of the miss and false alarm statistics computed over the stories in the system-defined topics that correspond to the reference target topics. Considering R and H as sets of stories, and using set operation notation, topic detection miss and false alarm probabilities are computed as:

$$P_{Miss} \;=\; \sum_R |R - H(R)| \Big/ \sum_R |R|$$
$$P_{FalseAlarm} \;=\; \sum_R |H(R) - R| \Big/ \sum_R |S - R|$$

A topic detection cost function, C_{Det}, will be used to combine the miss and false alarm probabilities into a single evaluation score. C_{Det} will serve as a bottom-line performance measure, to help in directing research effort and measuring research progress:

$$C_{Det} \;=\; C_{Miss} \cdot P_{Miss} \cdot P_{topic} \;+\; C_{FalseAlarm} \cdot P_{FalseAlarm} \cdot (1 - P_{topic})$$

In calculating performance, those stories tagged as BRIEF for the target topic will not be included in the correspondence sets nor the error tally. Stories tagged YES for more than one topic will be excluded from scoring altogether.

Evaluation of topic detection may be conducted under a total of up to 18 different conditions. This number is the product of 3 source conditions, 2 boundary conditions, and 3 deferral periods:

- ♦ Three source conditions:
  - ◊ **newswire text** and a **manual transcription** of the audio sources
  - ◊ **newswire text** and an **automatic transcription** of the audio sources
  - ◊ **newswire text** and the **sampled data signal** representing the audio sources[15]
- ♦ Two story boundaries conditions:
  Given **Not given**
- ♦ Three different decision deferral periods (in terms of # of source files, including the one being processed):
  1 **10** **100**

Each research site is encouraged to study as many of these conditions as may be productively done. However, all sites that perform the detection evaluation are required to perform at least one evaluation under the default conditions. The default conditions are:

- ◊ source – **newswire text** and an **automatic transcription** of the audio
- ◊ boundaries – **Given**
- ◊ deferral period – **10** source files

[15] Use of the sampled data signal allows simultaneous use of derivative forms of it, including automatically generated transcriptions.